Music Genre Classification with Digital Signal Processing Methods

Tarık Aytek, Tuğrul Alp Özbucak
Computer Engineering Department
Yildız Teknik University, 34220 Istanbul, Turkey
{tarik.aytek, alp.ozbucak}@std.yildiz.edu.tr

***Özetçe* —Bu çalışmanın ana amacı Müzik Türlerini Sınıflandırırken, Makine Öğrenmesi adımından önce verinin Gözetimsiz Kümeleme yöntemlerini ile belli miktarlarda kümelenip, sonuçları nasıl etkilediğini görmektir. Sonuçlarda görülmüştür ki, Gözetimsiz Kümeleme sonuçları sadece daha kötüleştirmektedir. Başarı düşme oranı normalde en başarılı olarak çıkan K-En Yakın Komşu algoritması için en şiddetli olarak görülmüştür.**

Anahtar Kelimeler—Gözetimsiz Kümeleme, Makine Öğrenmesi, Müzik Tür Sınıflandırma.

***Abstract*—The purpose of this paper is to see the effects of Unsupervised Clustering Algorithm's effects on success of Music Genre Classification when applied before the Machine Learning step. The results point towards the fact Unsupervised Clustering only decreases the success ratio. Some models are hit more severely than others, notably K-Nearest-Neighbor classifier.**

Keywords—Unsupervised Clustering, Machine Learning, Music Genre Classification.

I. Introduction

Music is everywhere within our lives and will remain to be as it is a fundamental way to appeal to people. But to use music in this age of information where data is bigger than we can imagine and analyze what music truly is, what makes it different from sounds we hear we need to find methods. That's the reason why Music Information Retrival (MIR) is so important [1]. To analyze music, first we have to extract it's features and apply methods to it to understand what it is. One such field is Music Genre Classification. It is needed to automatically figure out a song's genre so there may be further actions taken based on that. But it is not a perfected field as there are still improvements to be made and topics to be researched about.

To that end, this study researches the effects of using Unsupervised Clustering Algorithms on extracted features of the GTZAN data set [2]. The reason it's only Unsupervised is not having found many studies that use such methods. Feature extraction is primarily made with the help of Python and Librosa library [3]. Printing the values is done with the help of Pandas [4] and Machine Learning Algorithms are used from the Scikit library [5].

This paper is broken down to parts as following. Analyzing relevant studies in Section II. Feature extracting methods, Clustering Algorithms and Machine Learning Methods in Section III. executing in Section IV and analyzing results in Section V.

II. Related Works

In study [6], as features tempo spectrum, zero crossing rate, Mel-Frequency Cepstral Coefficients and Power Spectrum is used. For Clustering methods such as Neural Networks, Gaussian Mixture Model, Hidden Markov Model and Support Vector Machines are used. Results are as in Table 1. While the success rate is very high, the amount of features used and data set is limited.

	SVM	NN	GMM	HMM
Error Rate	6.86%	20.57%	12.31%	11.94%

Table 1 Clustering Results [6]

In this study [7], which is the most cited on MIR, the timbre texture features Spectral Center, Spectral Roll-off, Spectral Flow, Time Domain Zero Crossings, MFCC, Analysis and Texture Windows, Low Energy Feature were used and the mean and variance of these features were used as a vector. In addition, rhythmic and pitch features are also used for comparison. HMM and KNN were used for classification. In the study, it was observed that the success rate increased by %7 after 3 clustering, while the success rate was close to %81 with 2 clusterings in Classical music with the GMM method. With KNN, on the other hand, the success rates decreased as the number of clusters increased. As a shortcoming of this study, although it takes as many features as possible, it still has a low success rate. At the same time, clustering methods are very limited.

In another study, MFCC, Spectral Center, Spectral Roll-off, Spectral Flow, Zero Crossing and Low Energy were used as timbre texture properties. Tempo, rhythm and time measure were taken as rhythmic content features. Curtain features are also used. SVM, KNN, GMM, LDA were used as clustering methods. But the main point of the study is the DWCHs taken as attributes. Pointing to the previous study, they stated that they chose this new path because they saw the success rate as only %61 as a shortcoming. It has been observed that success rates are up to %80 in general, and up to %98 in specific genres [8]. In DWCHs, SVM2 performed %3.6 better than SVM1.

In our last example [9], which uses the GTZAN dataset and compares itself with almost previous studies mentioned, various previously unused timbre and temporal features are used on top of all the previously mentioned features and classification methods. This data is obtained through the

standard called MuVar, which is the combination of Mean and Variance; A standard called MuCov was obtained by combining the upper triangle of the mean and covariance matrix. By using SRC and SVM, the success rate was found to be over 90

III. Method Explanation

A. Feature Extraction Methods

1) Chroma STFT: Shows the amplitude of the twelve notes through analyzing the signal with Short Time Fourier Transform.

2) Chroma CQT: Used to overcome insufficient frequence resolution in STFT at lower frequencies. Like Discrete Fourier Transform it calculates frequency coefficients, but on a logarithmic scale.

3) Mel-Frequency Cepstral Coefficients : Mel-Frequency Cepstral Coefficients (MFCCs), are elements that are part of Mel-Frequency Cepstrum (MFC). In MFC frequency bands are equally spaced, which is more in line with human hearing. MFCCs are mostly used in Speech Recognition.

4) Spectral Centroid: Indicates where the gravital center of the spectrum is. Is also called as "brightness of a sound."

5) Spectral Bandwith: The standart deviation of the power spectrum near the Spectral Centroid.

6) Spectral Contrast: Amplitude difference of high and low parts of the spectrum.

7) Spectral Rolloff: The frequency value where below it rests the %85 or %95 of total spectral energy.

B. Clustering Algorithms

1) K-Means: It is targeted for there to be "k" amount of sides and each side have a center. First, each data is assigned to a random side, then the average is taken and the side centers are recalculated to find the situation that satisfies the condition. This continues until the cluster centers do not change [10].

2) Gaussian Mean Measure: Initially, each data is its own set. In each iteration, it continues to grow by adding the nearby center to itself. If the stop condition is not set, a single cluster is formed as the clusters will eventually merge [10].

3) Hierarchial Clustering: There are two types. In the top-down method, single data is divided according to distances. It is not used much because it requires the data to be in the dendogram structure first. In the bottom-up method, like GMM, each data is considered as a cluster and combined and continued [11].

C. Machine Learning Methods

1) Logistic Regression: Predicts the chance of an event happening, based on the variables in the dataset that are accepted to be unrelated to each other.

2) Random Forest: It is an ensemble method. It combines guesses of several base estimators from a given learning algorithm to make it more usable in many situations. In random forests each tree in the ensemble is built from a sample drawn with replacement from the training set. When splitting a node while creating a tree, the best split is derived from all available features.

3) K-Nearest Neighbor: The closest data node to the current center is picked and added, usually Euclidean Distance is used. Data should be normalized beforehand to avoid scale differences. To make nearer neighbors more important their weights can be given as 1/d where d is the distance between.

4) Support Vector Classifier: Given a training set with each data marked as being part of one of the two groups, a Support Vector Machine training algorithm builds a model that adds the new data to it's correct deemed group. It is a non-probabilistic binary linear classifier. Support Vector Machines maps training examples to points in space to maximize the distance between two groups. The new data is put on its place in this map and then related to the group depending on it's position.

Linear SVC is a variant that only accepts linear kernel and is specialized for that purpose.

5) Gaussians Naive Bayes: A probabilistic classification algorithm based on applying Bayes' theorem that assumes data are independent. By independent it is meant, one data point's existence doesn't influence the other's.

6) Extreme Gradient Boosting: XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It is normally used to train gradient-boosted decision trees and other gradient boosted models

It also has a Random Forest variant based on it's own principle. XGBoost Random Forests work with the same principle but with a different training algorithm.

7) Stochastic Gradient Descent: Stochastic Gradient Descent (SGD) is an iterative method for optimizing an objective function with suitable smoothness properties. It replaces the actual gradient derived from the data set by an estimate of a random subset in the data.

8) Decision Tree: A non-parametric supervised learning algorithm. It puts data in its correct place by asking simple questions in a chain, like a tree.

9) Linear Discriminant Analysis: A classifier with a linear decision boundary, generated by fitting class conditional densities to the data and using Bayes' rule. The model fits a Gaussian density to each class, assuming that all classes share the same covariance matrix.

10) Gradient Boosting: Gradient Boosted Decision Trees (GBDT) is a generalization of boosting to arbitrary differentiable loss functions. It's an ensemble of weak prediction models, which are typically decision trees. When such a tree is the weak learner, the algorithm becomes a gradient-boosting algorithm.

11)AdaBoost: Adaboost works by trying to fit series of weak learners on constantly modified versions of the data set. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights to each of the training samples. Each weak learner learns from the mistakes of others and improves.

IV. The Study

GTZAN data set is first broken down to equal, 5 second length parts to amplify the strong parts and weaken the unwanted parts of the signal. Then, from this newly generated data set the Chroma STFT, Chroma CQT, MFCC-20, Spectral Centroid, Spectral Bandwith, Spectral Contract and Spectral Roll-off features are extracted using Librosa library. Then these feature values are subjected to Unsupervised Clustering Algorithms such as K-Means, Gaussian Mixture Method and Hierarchial Clustering by two and three centers.

After clustering, the mean, standart deviation, minimum, maximum, median, mod, skew and kurtosis of no cluster, KNN-2, KNN-3, GMM-2, GMM-3, Hier-2 and Hier-3 is calculated. Using these stats models are trained and tested. The Machine Learning Algorithms used are K-Nearest Neighbor(K-NN), Support Vector Machine(SVM), Random Forest(RF), Gaussian Naive Bayes(GNB), XGBoost Gradient Boosting(XGBoost), Stochastic Gradient Descent(SGD), Decision Tree, XGBoost Random Forest, Gradient Boosting(GB), AdaBoost and Linear Support Vector Machine(Linear SVM). After this results are obtained through Pandas library.

V. Examination of Results

Classifier	Accuracy	Precision	F1
KNN	0.882	0.835	0.835
XGBoost	0.860	0.861	0.860
Random Forest	0.837	0.835	0.835
Support Vector	0.803	0.802	0.801
Gradient Boosting	0.804	0.805	0.804
LSV	0.782	0.780	0.780
Logistic Regression	0.757	0.756	0.755
XGBoost RF	0.753	0.749	0.748
LDA	0.747	0.753	0.748
SGD	0.677	0.734	0.669
Decision Tree	0.603	0.605	0.603
GNB	0.546	0.550	0.528
AdaBoost	0.354	0.358	0.301

Table 2 No Cluster

First of all, it has been shown again that the K-NN, XGBoost, RF, SVM and GB methods give more than %80 success without any clustering attached.

For most modelling classifiers there has not been any notable variance between success classifiers. The only notable classifier would be SGD, where Precision was almost always 0.05 higher than all other metrics.

	KNN	XGBoost	RF	SVM	GB
No Cluster	0.882	0.860	0.837	0.803	0.804
K-Means -2	0.869	0.844	0.829	0.775	0.797
K-Means -3	0.695	0.833	0.810	0.747	0.794
GMM -2	0.796	0.823	0.805	0.751	0.773
GMM -3	0.642	0.818	0.779	0.716	0.772
Hierarch -2	0.753	0.813	0.770	0.744	0.763
Hierarch -3	0.584	0.798	0.762	0.717	0.756

Table 3 Overall Meaningful Result Comparison

When we look at the table 3 which shows a summary of useful results, according to the clustering method, it can be seen that the success rates decrease in all clustering methods. Again, this decrease can generally be seen least in K-Means, on average in GMM, and most severely in Hierarchial Clustering.

It has been seen that the KNN method, which gives the highest result when clustering is not done, is affected by KNN the least and Hierarchial Clustering the most. Compared to no clustering, Hier 2 clustering showed %10 less success. It was observed that the success of KNN decreased by more than %15 when the clustering algorithms were performed in triples, whereas in Hier-3, there was an almost %30 decrease in success compared to the clusterless state. In Kmeans-2, although Accuracy decreased, all other metrics increased by %3.

In the end it is found that using Unsupervised Clustering Algorithms before Modelling Phase does more harm than good. But it is also seen some models are more resistant to it than others, like how the natural winner K-NN method is extremely weak against it. It is also seen K-Means decreases the success rate the least, while Hierarchial Clustering decreases it the most. But in the end they still do decrease it and won't be any meaningful use in Music Genre Classification.

References

[1] J. S. Downie, "Music information retrieval," *Annual review of information science and technology*, vol. 37, no. 1, pp. 295–340, 2003.

[2] A. Olteanu. (2020) Gtzan dataset - music genre classification. [Online]. Available: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

[3] (2022, 02) Librosa:0.9.1. [Online]. Available: https://zenodo.org/record/6097378

[4] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[6] C. Xu, N. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 5, 2003, pp. V–429.

[7] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[8] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," 01 2003, pp. 282–289.

[9] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303–319, 2011.

[10] S. R. A. Ahmed, I. Al Barazanchi, Z. A. Jaaz, and H. R. Abdulshaheed, "Clustering algorithms subjected to k-mean and gaussian mixture model on multidimensional data set," *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 2, pp. 448–457, 2019.

[11] F. Nielsen, *Hierarchical Clustering*. Cham: Springer International Publishing, 2016, pp. 195–211. [Online]. Available: https://doi.org/10.1007/978-3-319-21903-5$_8$